CHAPTER

14

SERVER MONITORING AND OPTIMIZATION

After reading this chapter and completing the exercises you will be able to:

- Establish monitoring benchmarks
- Monitor server services, logged-on users, and server functions
- Use Task Manager to monitor processes and performance data
- Use the System Monitor to monitor page file, memory, processor, disk, and other critical server performance functions and to tune these functions as needed
- Set up performance logs and alerts for monitoring
- Identify key system elements to monitor for problems

Monitoring a server can mean the difference between being caught off guard when a problem strikes, and anticipating and correcting an impending problem before the users notice. For example, a failing disk adapter may be diagnosed and replaced before it fails. In another situation, you may implement a popular new application that requires more server memory than anticipated, a potential problem that you can monitor and correct before users realize there is a problem. Server monitoring enables you to be proactive in maintaining fast server response, and it gives you the tools to quickly find and resolve problems after they strike.

In this chapter, you learn about the Windows 2000 Server services that keep everyday functions going and how to monitor those services. You also learn to use the Task Manager to monitor the server and help you optimize its performance. The System Monitor is the most powerful monitoring tool, which you learn to use in a multitude of ways for tracking system performance and determining how to optimize server functions. You also learn how to create logs for storing performance information and how to set up alerts to warn you of a problem as soon as it occurs.

MONITORING THE SERVER

Server monitoring accomplishes several purposes. One reason to monitor is to become familiar with your server's performance so you know how to interpret a problem. It may be difficult to diagnose a problem or determine if there is a resource shortage unless you first know what performance is typical for your server. Other reasons to monitor are to prevent problems before they occur and to diagnose existing problems to resolve them. Monitoring enables you to pinpoint problems and identify solutions, for example by tracking disk errors and replacing a hard disk before it fails. Table 14-1 shows some typical performance areas that play a significant role in a server's response and that can be monitored through the tools included with Windows 2000 Server.

Table 14-1 Server Activities to Monitor

Monitoring Area	Factors Causing the Problem
Server services	Hung or stopped service, or one using a high percentage of CPU resources
Logged-on users	Number of users logged on and types of resources they are accessing
Software	Server resources used by software packages
Paging	Page file sizing and performance
RAM	Memory shortage or damaged memory
CPU	CPU utilization and performance
Hard disk	Disk performance, capacity, and errors
Caching	Cache allocation and performance

Establishing Server Benchmarks

The most important way to get to know your server is to use monitoring tools to establish normal server performance characteristics. This is a process that involves establishing benchmarks. **Benchmarks**, or baselines, provide a basis for comparing data collected during problem situations with data showing normal performance conditions. This creates a way to diagnose problems and identify components that need to be upgraded. Benchmarks are acquired in the following ways:

- By generating statistics about CPU, disk, memory, and I/O with no users on the system, to establish a baseline for comparison to more active periods. Keep spreadsheets or databases and print performance charts of this information.
- By using performance monitoring to establish slow, average, and peak periods. Keep records on these periods.
- By gathering performance statistics each time a new software application is installed, on slow, average, and peak periods during its use.
- By establishing benchmarks to track growth in the use of servers, such as increases in users, increases in software, and increases in the average amount of time users are on the system.

The best way to get a feel for a server's performance is to gather benchmarks and then to frequently monitor server performance after you have the benchmark data. Performance indicators can be confusing at first, so the more time you spend observing them, the better you'll understand them. For example, viewing the CPU utilization on a server the first few times does not tell you much, but viewing it over a period of two or three months, noting slow and peak periods, helps you develop knowledge about how CPU demand varies for that server.

Using Windows 2000 Server Services

Windows 2000 Server automatically starts a range of system services that run in the background as the server is running and that should be monitored periodically. Many are default services that are automatically installed when you first install Windows 2000 Server. Other services can be installed or added when you install additional Windows components from the Windows 2000 Server CD-ROM (such as DNS services) or from independent software sources. Some are automatically started when the server boots, and others are started manually as needed. There are several default services that provide for messaging, logging, scheduling, server, and printer activities. If the server is having performance problems, you have the option to check the services and determine if one is stopped or possibly hung. You also have the option to stop an unused service to ease the server load. The default services that are automatically started are summarized in Table 14-2.

Table 14-2 Windows 2000 Server Default Services

Service	Description
Alerter	Sends notification of alerts or problems on the server to users designated by the network administrator
Computer Browser	Keeps a listing of computers and domain resources to be accessed (see the Note following this table)
EventLog	Enables server events to be logged for later review or diagnosis, in case problems occur
File Replication Service	Replicates the Active Directory elements on multiple domain controllers, when the Active Directory is installed
Intersite Messaging	Transfers messages between different Windows 2000 Server sites
IPSEC Policy Agent	Enables IPSec security and management
Kerberos Key Distribution Center	Enables Kerberos authentication and enables the server as a center from which to issue Kerberos security keys and tickets
Licensing Logging Service	Enables the monitoring of server licensing and other licensing
Logical Disk Manager	Monitors for disk problems, such as a disk that is nearly full
Messenger	Handles messages sent for administrative purposes
Net Logon	Maintains logon services such as verifying users who are logging on to the server or a domain
Plug and Play	Enables automatic detection and installation of new hardware devices or devices that have changed

Table 14-2 Windows 2000 Server Default Services (continued)

Service	Description
Print Spooler	Enables print spooling
Protected Storage	Enables data and services to be stored and protected by using private key authentication
Remote Procedure Call (RPC)	Provides remote procedure call services
Remote Procedure Call (RPC) Locator	Used in communications with clients using remote procedure calls to locate available programs to run
Remote Registry Service	Enables the Registry to be managed remotely
Removable Storage	Enables management of removable storage media, such as tapes, CD-RWs, and Zip and Jaz drives
RunAs Service	Used to run programs via an account that is different from the one currently logged on, such as running a program as Administrator from a computer logged on to another user's account (so that an Administrator does not have to log off a user).
Security Accounts Manager	Keeps information about user accounts and their related security setup
Server	A critical service that supports shared objects, logon services, print services, and remote procedure calls
System Event Notification	Enables the detection and reporting of important system events, such as a hardware or network problem
Task Scheduler	Used to start a program at a specified time and works with the software Task Scheduler
TCP/IP NetBIOS Helper Service	Activated when TCP/IP is installed and used to enable NetBIOS name resolution and NetBIOS network transport
Uninterruptible Power Supply	Used with a UPS to coordinate supplying power to the server during power failures
Windows Time	Enables updating the clock
Workstation	Enables network communications and access by clients over the network

Computers on a network can be viewed within Windows 2000 by means of the Computer Browser service. The service is used by tools such as My Computer, the MMC, and others to view computers. Any Windows 3.1, 3.11, 95, 98, NT, or 2000 computer configured for network access is a part of the Browser system. A Windows 2000 server is selected through the Computer Browser service to be the **master browser**, keeping the main list of logged-on computers, while the other browsers play a support role, such as that of **backup browsers**, which maintain a copy of the list in case the master browser is offline. There are also potential browsers that can be promoted to backup browsers, if more are needed, and nonbrowsers, which are computers that do not maintain a list. The Server and Workstation services depend upon the Computer Browser service.

If a LAN Manager client is attached to the network, it will not receive a browse list from Windows 2000 Server or Windows NT Server, unless the server is configured to send the list. To configure Windows 2000 Server, open the Network and Dial-up Connections tool, right-click Local Area Connection, click Properties, double-click File and Printer Sharing for Microsoft Networks, and click Make browser broadcasts to LAN manager 2.x clients (see Chapter 7). (To configure Windows NT Server 4.0 for LAN Manager 2.x browser broadcasts, open the Control Panel, double-click the Network icon, and click the Services tab.)

Monitoring Server Services

As you have already learned in earlier chapters, Windows 2000 Server services can be viewed and managed from the Computer Management tool, which is opened from the Administrative Tools menu (click Start, point to Programs, point to Administrative Tools, and click Computer Management). Two other ways to manage services are from the Administrative Tools menu Services option or by using the MMC Services snap-in. For example, if you open Services from the Administrative Tools menu, a console window opens, as shown in Figure 14-1. Services are displayed in the right pane of the window, which contains five columns. The Name column shows services listed alphabetically. A short description of each service is provided in the Description column. The Status column indicates the condition of the service as follows:

- *Started* shows that the service is running.
- *Paused* means that the service is started, but is on hold to the users.
- A blank means that the service is halted or has not been started.

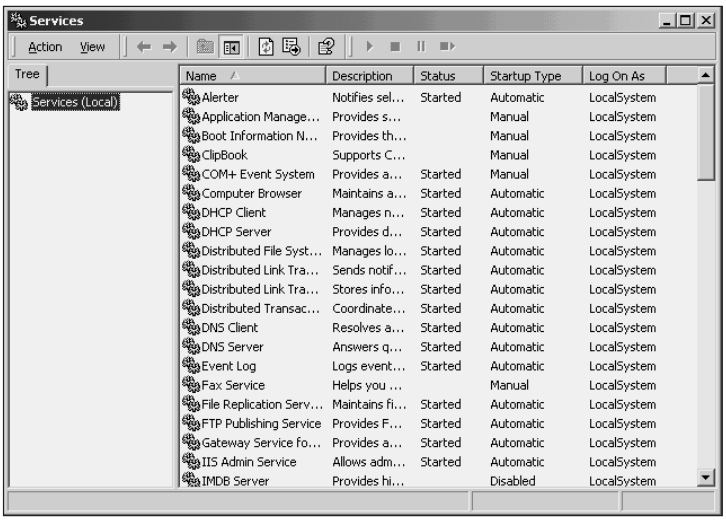

Figure 14-1 Monitoring server services

The Startup Type column shows how a service is started. Most services are started automatically when the server is booted. Some services are started manually because they may not be needed until a given time. In Figure 14-1, the ClipBook service is not currently running and is set to start manually when needed by the administrator. The ClipBook service is used to enable remote users to access ClipBook contents on the server. The Log On As column shows information about where the service is running, such as on the local computer system, which is usually the case, or from a specialized area, as with a program service that runs through the Administrator account.

When you use the Services MMC snap-in, you can monitor services on another Windows 2000 or NT server, or even on a computer running Windows 2000 Professional or Windows NT Workstation. To access another computer's services, start MMC, click Console, click Add/Remove Snap-in, click Add, double-click Services, click Another computer, specify the computer's name or click Browse to find it, click Finish, click Close, and then click OK.

When you experience a problem on a server that is associated with a service, check the status of the service to make sure that it is started or set to start automatically. You can start, stop, pause, resume, or restart a service by right-clicking it and clicking any of these options. For example, occasionally a service does not start properly when the server is booted or hangs while the server is running, such as the Print Spooler service. The Services tool provides a way to monitor this situation. Even if the Print Spooler shows that it is started and you determine that you want to restart it, right-click the service and click Restart.

In another example, consider a message at the server that the Server service is suspended because of a problem, and you cannot log on to the server from the workstation in your office. When you log on to the server console as Administrator, the logon process takes four minutes, and when you click the Printers folder for a shared printer, the folder hangs for a minute and aborts with a message that the Server service is suspended. One way to address the problem is to open the Services tool to view the currently started services and to check the Server service to see if it is stopped or paused. If it is stopped or paused, you can start it by right-clicking the Server service and clicking Start.

Use the stop option carefully, because some services are linked to others. Stopping one service will stop the others that depend on it. For instance, stopping the Workstation service affects the Alerter, Computer Browser, Distributed File System, Messenger, Net Logon, and Remote Procedure Call services. The system gives you a warning when other services are affected by stopping a particular service.

You can check dependencies by double-clicking a service and clicking the Dependencies tab (see Figure 14-2). (Try starting and stopping a service, as well as viewing its dependencies, in Hands-on Project 14-1.)

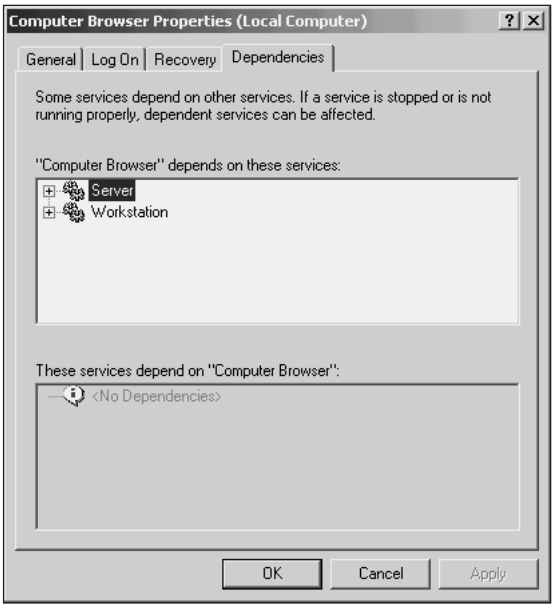

Figure 14-2 Service properties

Many services are linked to the Server and Workstation services, including logged-on users. If it is necessary to stop one of these services—for example, to diagnose a problem—give the users advance warning or stop it after work hours.

Pausing a service takes it offline to be used only by Administrators or Server Operators. For example, if the ClipBook service is sending error messages to users, you can pause the service so it is only available to the Administrator for testing until the problem is resolved. A paused service is restarted by right-clicking it and clicking Restart.

Another way to manage a service is to double-click it to view that service's properties (see Figure 14-2). For example, you can set a service to start automatically by double-clicking the service, accessing the General tab, and setting the Startup Type box to automatic. The function of each properties tab is presented in Table 14-3.

Table 14-3 Services Properties Tabs

Properties Tab	Description
General	Displays general information about the service, enables you to determine whether to start the service automatically, start it manually, or disable it; you can also start, stop, pause, and resume services
Log On	Enables you to specify the account that the service uses to log on, which is normally the local system account
Recovery	Enables you to specify how the computer will respond if the service fails, for example by automatically restarting the service
Dependencies	Displays the services that depend on a particular service and the services on which a particular service depends

Monitoring Logged-on Users and Resource Use

Network administrators frequently monitor the number of users who are accessing a server, for several reasons. One is to develop an indication of how many users are typically logged on at given points in time, which provides information about normal user load. Also, if a problem develops and the server needs to be shut down, the administrator can determine when the shutdown will have the least impact. Another reason is to be aware of security or misuse problems, such as an account that is in use when the owner is not at his or her workstation. On large networks, it is a good idea to frequently check the number of users on a server. An especially popular server may need to be upgraded as more users log on for extended periods.

To view logged-on users, start the Computer Management tool from the Administrative Tools menu, double-click objects in the tree as necessary to view Computer Management (Local), System Tools, Shared Folders, and Sessions. Click Sessions. The right-hand pane shows the users who are connected with active sessions, the operating system used at the client, the number of files each user has open, the connected time, the idle time, and whether the user is logged on as a guest. Depending on how it is connected, a single client may have two or more connections, such as a network connection for the computer and a connection for the user account. If you need to close a user's session—for example, if the session is hung and stays active after the user has logged off—right-click the user and click Close Session.

You can also view information about which shared folders are being accessed by users when you click Shares instead of Sessions in the console under Shared Folders (see Figure 14-3). The # *Client Redirections* column shows the number of clients using shares. Notice in Figure 14-3 that some shares, such as C$ and print$, are set up by default as hidden ($ after the share name hides the share on the network). The print$ share enables you to view the number of clients currently using the server as a print server. You can stop sharing by right-clicking any shared file and clicking Stop Sharing. Also, you can view which files are in use by clicking Open Files under Shared Folders. The right-hand pane shows the named pipes, number of open files, file locks, and permissions mode, such as Read+Write. The open file information shows if a file is in use. A **file lock** means that no one else can update a specified file, and **named pipes** are open communication links between two processes on the server or between the server and a client. This information is useful, for example, if one user has a file in use and

another user cannot update the file because it is locked. You can check the lock information and inform the user who wants to update that the file is already in use by someone else. Or, if you need to close the connection that has the file locked, so that the second user can perform an update, right-click the connection you want to close and click Close Open File (try Hands-on Project 14-2).

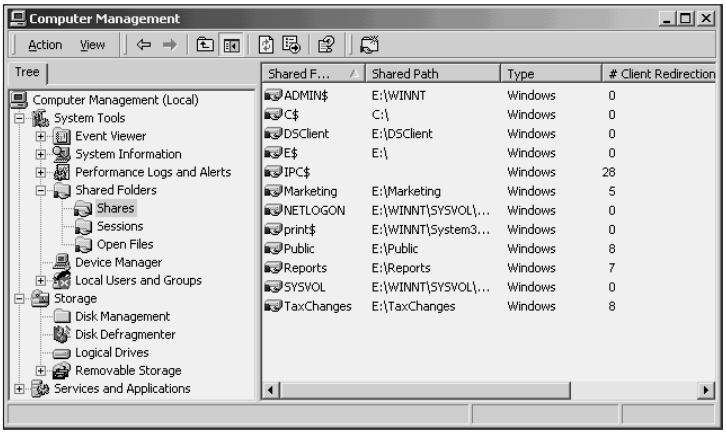

Figure 14-3 Shared resources

Monitoring Applications with Task Manager

Not all software applications are designed equally. Some have an extremely inefficient design that places unnecessary demand on the server. For example, an application may generate excessive network traffic by transporting more data than it needs between the server and the client. Another source of demand is reports that run against a database, requiring the system to read all of the records in the database instead of the limited few needed for each report. An inefficient program is often signaled by high CPU, memory, or disk utilization each time the program runs.

Some applications advertised to be client/server based are not truly designed according to these standards and can be inefficient when using server and network resources. For example, client/server design is intended to efficiently spread the workload between server and clients, but some applications place excessive work on the server or use poor database design at the server. Check on actual performance before you buy.

You can use Task Manager to view applications running on the server by pressing Ctrl+Alt+Del while logged on as Administrator or with Administrative privileges. After pressing this key combination, you will see the following options:

- *Lock Computer:* Secures the Windows 2000 Server console from access
- *Change Password:* Used to change the password of the account currently logged on
- *Log Off:* Logs off the account that is currently logged on
- *Task Manager:* Used to view information about tasks and services currently running
- *Shut Down:* Shuts down the server
- *Cancel:* Returns to the Windows 2000 desktop

An alternate way to start Task Manager is to right-click an open space on the taskbar and click Task Manager.

Click the Task Manager button, which displays a dialog box with three tabs: Applications, Processes, and Performance. The Applications tab, shown in Figure 14-4, displays all of the software applications running from the server console, including 16-bit applications. Any of the applications can be stopped by highlighting it and clicking the End Task button. If an application is hung (no longer responding to user input), you can press End Task to release more resources for the server. The Switch To button brings the highlighted application to the front so you can work in it, and the New Task button enables you to start another application at the console, using the Run option, which is the same option that you would access from the Start button. The status bar at the bottom of the screen shows information about the total number of processes, the CPU usage, and minimum/maximum available memory. For example, in Figure 14-4, there are 46 total processes running, using 6% of the CPU and 159060K out of 310304K total memory. As used in the context of the Task Manager, a **process** is an executable program or one or more executable programs that run from a main program. For example, when you click Help from the Microsoft Excel menu bar, the Winhlp32.exe process runs along with Excel.exe.

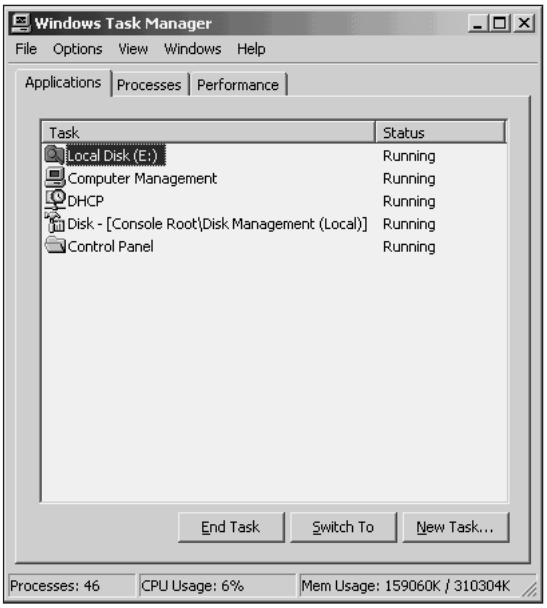

Figure 14-4 Monitoring started tasks

As you will learn as you monitor a server, even if CPU utilization goes to 100% this is not a cause for immediate concern, because it may mean that a process is simply using the CPU very efficiently. However, if utilization frequently stays at 100% for several minutes instead of several seconds, this is cause for concern and may indicate a software or hardware problem.

If you right-click a task or highlight a task and right-click the heading in the Task column or Status column, several options appear in a shortcut menu, as follows:

- *Switch To:* Takes you into the highlighted program
- *Bring to Front:* Maximizes and brings the highlighted program to the front, but leaves you in the Task Manager
- *Minimize:* Causes the program to be minimized
- *Maximize:* Causes the program to be maximized, but leaves you in the Task Manager
- *End Task:* Stops the highlighted program
- *Go To Process:* Takes you to the Processes tab and highlights the process associated with the program

The Processes tab lists the processes in use by all running applications. If you need to stop a process, you can stop it from this screen by highlighting it and clicking End Process. Also, the Processes tab shows information about each started process, as summarized in Table 14-4.

The Processes tab lists the processes in use by all running applications. When a process is started or while it is running, it may start other processes, all of which compose the **process tree**. The additional processes in the process tree are displayed on the Processes tab so that they are indented under a main process. For example, when you start a 16-bit process, such as one called Image.exe in Windows 2000 Server, that causes two other processes to start, the virtual DOS machine (Ntvdm.exe, see Chapter 1) and Windows on Windows (Wowexec.exe). Because Image.exe is a 16-bit process, it must directly start Ntvdm.exe, which is a 32-bit process started by Windows 2000 Server in order to run a 16-bit process within it. Also, Image.exe indirectly starts Wowexec.exe ,which simulates a 16-bit window in Windows 2000. All three processes compose the process tree, and the Processes tab shows Ntvdm.exe as the main process, with image.exe and Wowexec.exe running under it.

If you need to stop a process, you can stop it from the Processes tab by highlighting the process and clicking End Process. To stop all direct and indirect processes in a process tree, right-click the main process (the one above the indented processes in the tab) and click End Process Tree. Also, the Processes tab shows information about each started process, as summarized in Table 14-4.

Table 14-4 Task Manager Information on Processes

Process Information	Description
Image Name	The process name, such as winword.exe for Microsoft Word
PID	The process identifier (PID), which is an identification number assigned to the process so the operating system can track information on it
CPU	The percentage of the CPU resources used by the process
CPU Time	The amount of CPU time used by that process from the time the process started
Mem Usage	The amount of memory the process is using

Table 14-4 lists only the default information that is displayed on the Processes tab. You can change the display to view other information, such as page faults, base priority class, and threads (all described later in this chapter) by clicking the View Menu and then clicking Select Columns.

For example, if you suspect that a program is causing a bottleneck at the CPU, go to the Applications tab, right-click the program, and click Go To Process, which identifies the program's process on the Processes tab. Next, look in the CPU and CPU Time columns to see how much of the CPU that program's process is using. Also, check the figure in the Mem Usage column to see if the program is causing a memory bottleneck. If the program is using too many resources, such as 90% of the CPU, consider stopping it and discontinuing its use until you know the source of the problem.

Also, you can increase the priority of a process (or processes) in the list so that it has more CPU priority than what is set as its default. Suppose, for example, that you want to increase the priority for Windows Explorer, which is process Explorer.exe. To start, right-click

Explorer.exe, displaying a shortcut menu in which you can end the process, end the process tree (end that process and all subprocesses associated with it), or reset the priority. Click Set Priority to reset the priority (see Figure 14-5).

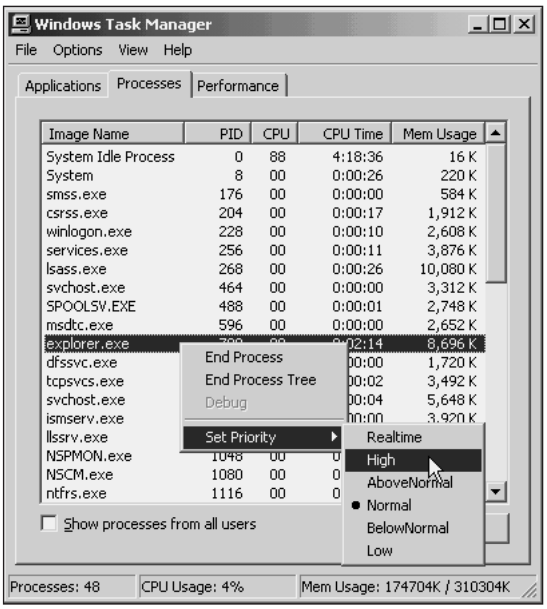

Figure 14-5 Resetting a process priority

Normally, the priority at which a process runs is set in the program code of the application, which is called the **base priority class**. If the base priority class is not set by the program, a normal (average) priority is set by the Windows 2000 Server operating system. The server administrator always has the option to set a different base priority. As shown in Figure 14-5, the administrator can change the priority to any of six options: Low, BelowNormal, Normal, AboveNormal, High, or Realtime. You might think of these processes as being on a continuum, with Normal as the midpoint, which is 0. Low is -2, BelowNormal is -1, AboveNormal is +1, or High is +2. Realtime is given an extra advantage at +15. For example, a Low priority means that if a process is waiting in a queue—for example, for processor time, disk access, or memory access—all processes with a higher priority will go first. The same is true for BelowNormal, except that processes with this priority will run before those set at Low, and so on. (Try Hands-on Project 14-3 to practice resetting a priority and stopping a task.)

Use the Realtime priority with great caution. If assigned to a process, that process may completely take over the server, preventing work by any other processes. For instance, you might want to assign a Realtime priority when you detect a disk drive that is about to fail and you want to give all resources over to the backup process so you can back up files before the disk fails.

The Performance tab shows vital CPU and memory performance information through bar charts, line graphs, and performance statistics (see Figure 14-6). The CPU Usage and MEM Usage bars show the current use of CPU and memory. To the right of each bar is a graph showing the immediate history statistics. The bottom of the Performance tab shows more detailed statistics, such as those for handles and threads, which are described in Table 14-5. A **handle** is a resource, such as a file, used by a program and having its own identification so the program is able to access it. **Threads** are blocks of code within a program.

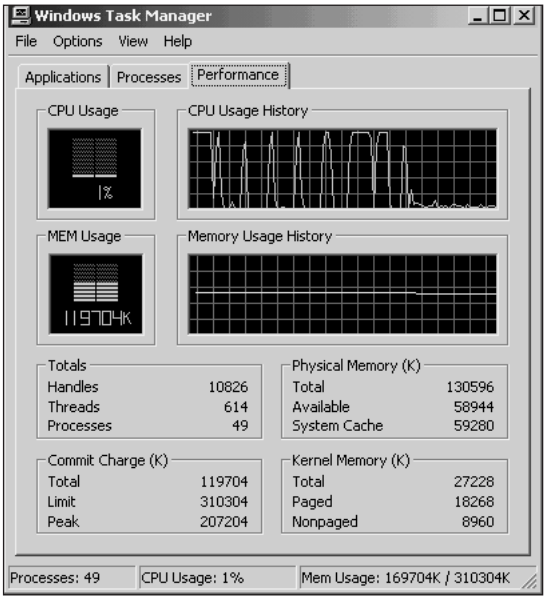

Figure 14-6 Performance data

Notice that in Figure 14-6 the Physical Memory Total is 130596K, but the Commit Charge Total is only 119704K, which is the setting for the initial page size. Also, notice that the Commit Charge Peak for this monitoring session is 207204K, which shows that the initial page size has already been exceeded. As discussed in Chapter 7, the initial page file size should be equal to 1.5 times the amount of installed RAM. Thus, the performance statistics show that the page file size may be set too low at under 195894K (130596 * 1.5) and that the server administrator may need to tune the virtual memory for this server. The page file is tuned by opening the Control Panel System icon, then clicking the Advanced tab, clicking the Performance Options button, and clicking the Change button.

Table 14-5 Task Manager Performance Statistics

Statistic	Description
Handles	The number of objects in use by all processes, such as open files
Threads	The number of code blocks in use, in which one program or process may be running one or more code blocks at a time
Processes	The number of processes that are active or sitting idle
Physical Memory Total	The amount of RAM installed in the computer
Physical Memory Available	The amount of RAM available to be used
System Memory File Cache	The amount of RAM used for file caching
Commit Charge Total	The size of virtual memory currently in use
Commit Charge Limit	The maximum virtual (disk) memory that can be allocated
Commit Charge Peak	The maximum virtual memory that has been used during the current Task Manager monitoring session
Kernel Memory Total	The amount of memory used by the operating system
Kernel Memory Paged	The amount of virtual memory used by the operating system
Kernel Memory Nonpaged	The amount of RAM memory used by the operating system

USING SYSTEM MONITOR

The most vital tool used to help detect and fix any problems on a Windows 2000 server is **System Monitor**. System Monitor is like a window into the inner workings of just about every aspect of the server, such as hard disks, memory, the processor, disk caching, a started process, and the page file. For example, you might monitor memory and page to determine if you have fully tuned the page file for satisfactory performance and to determine if you have adequate RAM for the server load.

System Monitor is opened from the Administrative Tools menu by clicking Performance to view the console screen, which offers two main choices. System Monitor is the top selection in the tree, and Performance Logs and Alerts (discussed later in this chapter) is the other option. Make sure that System Monitor is selected in the tree to view the screen as shown in Figure 14-7. The screen is in the chart mode, showing a grid that you use for graphing activities on the server. To begin tracking, you must select one or more objects to monitor. A System Monitor object may be memory, the processor, or another part of the computer. Table 14-6 lists the main computer system objects that can be monitored (in the next chapter, network objects are presented). Other objects are added as you add server services. For example, when you install IIS, more objects are added to monitor Internet Information Services, the HTTP Service, and FTP Server activity.

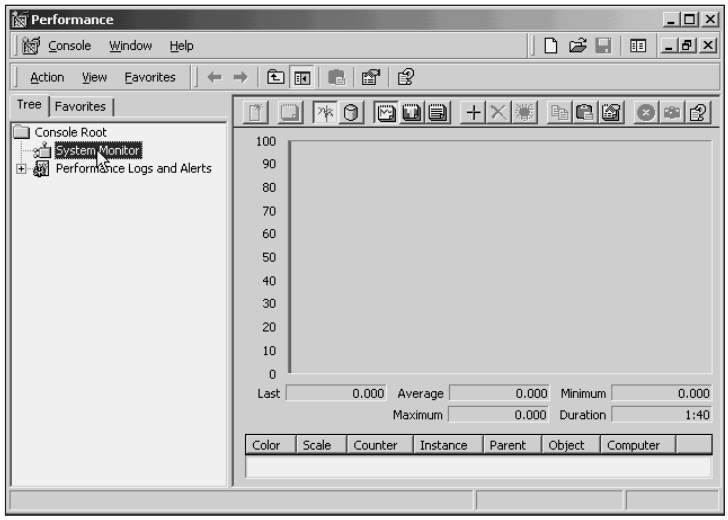

Figure 14-7 System Monitor

Table 14-6 System Monitor Objects

Object	Description
Active Server Pages	Monitors client requests, transactions, and sessions at the server
Browser	Tracks the activity of the browser service that enables My Network Places to communicate and exchange information with other computers on the network, for example tracking duplicate master browsers
Cache	Supplies performance information on data caching
Distributed Transaction Coordinator	Monitors distributed transactions processed through the server
Job Object	Monitors CPU, kernel mode, and user mode activity for a job object (a group of processes that are handled as one entity)
Job Object Details	Monitors detailed information about a job object, such as I/O activity, paging activity, and working set activity
Memory	Provides information about RAM use, such as percent of memory in use, amount of memory available to processes, caching, and paging
Objects	Tracks the activities of special objects, such as started processes and started threads
Paging File	Supplies data on page file performance, such as current usage and peak usage
Process	Supplies performance data on a specific process that is running
Processor	Tracks demands on the processor such as the percent in use, the number of requests from hardware components, percent in use by users, and the percent in use by the operating system

Table 14-6 System Monitor Objects (continued)

Object	Description
Redirector	Monitors network connection information, such as folder access requests from other computers on the network and information about workstations presently connected
Server	Tracks information about the Server service, such as number of bytes sent out and received, logon errors, and logged-off sessions
Server Work Queues	Provides information about active threads, bytes received from clients and sent to clients, length of the CPU's work queue, and rate at which the server is reading and writing data
System	Monitors file access, system calls, operating system activities, processes handled by the server, and processor queue length
Telephony	Monitors activity to telecommunications lines connected to the server
Thread	Tracks a specific thread running within a process, such as the processor time used by the thread

For each object, there are one or more counters that can be monitored. A **counter** is an indicator of a quantity of the object that can be measured in some unit, such as percent, rate per second, or peak value, depending on what is appropriate to the object. For example, the % Processor Time counter for the Processor object measures the percentage of processor time that is in use by non-idle processes. Pages/sec is an example of a counter for the Memory object that measures the number of pages written to or read from virtual memory per second. The processor is one of the common objects to monitor when a workstation or server is slow. Table 14-7 gives examples of some of the most frequently used counters for the Processor object (try Hands-on Project 14-4 to view objects and counters).

Table 14-7 Sample Processor Counters in System Monitor

Counter	Description
% DPC Time	Processor time used for deferred procedure calls, for example for hardware devices
% Interrupt Time	Time spent on hardware interrupts by the CPU
% Privileged Time	Time spent by the CPU for system activities in privileged mode, which is used for the operating system
% Processor Time	Time the CPU is busy on all non-idle activities
% User Time	Time spent by the CPU in user mode running software applications and system programs
Interrupts/sec	Number of device interrupts per second

Sometimes there are instances associated with a counter. An **instance** exists when there are different elements to monitor, such as individual processes when you use the Process object, or when a process contains multiple threads or runs subprocesses under it for the Thread object. Other examples are when there are two or more disks or multiple processors to monitor. In many cases, each instance is identified by a unique number for ease of monitoring.

System Monitor Options

System Monitor offers several buttons from which to operate it and to set up the display options. After the tool is opened, click the Add button (represented by a plus sign) on the button bar just above the tracking window (refer to Figure 14-7), to access the dialog box from which to select objects to monitor, counters, and instances (described in more detail in the following sections). You can monitor one or more objects at a time as a way to get a better understanding of how particular objects interact, for example by monitoring both memory and the processor. Also, you can monitor the same object using different combinations of counters. You stop monitoring by clicking the Delete button (represented by an X) on the button bar.

When you monitor objects, you can use one of three modes: chart, histogram, or report. A chart is a running line graph of the object that shows distinct peaks and valleys. For example, when you use the chart mode and monitor for different objects, each object is represented by a line with a unique color, such as red or green. A histogram is a running bar chart that shows each object as a bar in a different color. The report mode simply provides numbers on a screen, which you can capture to put in a report. Each of these options is set from a button on the button bar just above the tracking window, and the buttons are titled: View Chart, View Histogram, and View Report. You can change the view mode at any time by clicking the appropriate button. Figure 14-8 illustrates the use of the chart mode to monitor several counters.

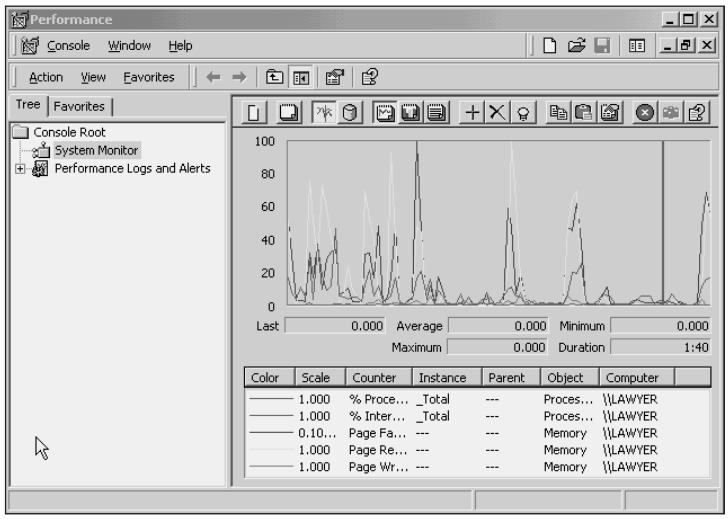

Figure 14-8 System Monitor chart mode

The System Monitor can also be set up to run from inside Microsoft Word so that you can create visual reports and print them using Word (or send them through e-mail). To run System Monitor in Word (see Figure 14-9):

1. Start Microsoft Word (Word 97 or higher).
2. Click the View menu, highlight Toolbars, and click Control Toolbox.

3. Click More Controls (the icon with the hammer and ruler).
4. Scroll to find System Monitor Control, and click that option.
5. Adjust the size of the figure display and move it to the desired location in the document.
6. Click the Exit button.
7. Use the options in System Monitor as you would normally, for example to select counters, objects, and instances and to select the monitor mode, such as charting (see the next section).
8. Use the Word commands in a normal fashion to format the document, save it, and print it.

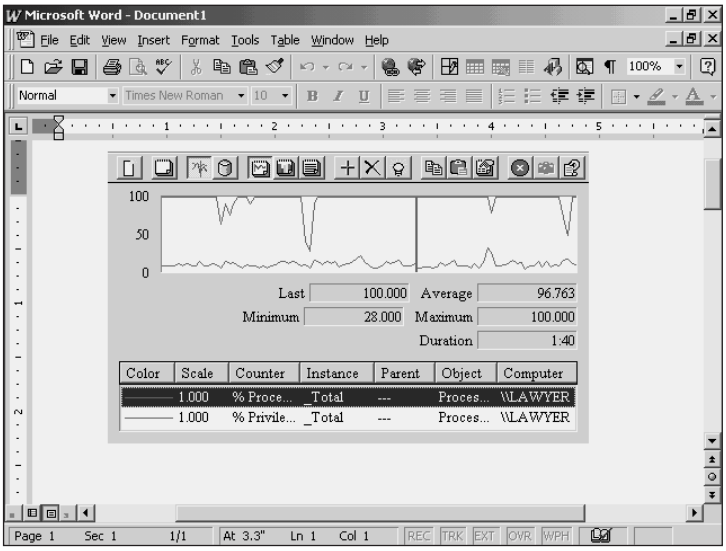

Figure 14-9 Running System Monitor inside Microsoft Word

Monitoring Page File and Memory Performance

To monitor the server's page file performance (recall that a page file is disk space reserved for use when memory requirements exceed the available RAM), click the Add button (plus sign) to bring up the Add Counters dialog box, in which you can select the computer and the objects to monitor. At the top of the box, select the computer to monitor by clicking either *Use local computer counters* for the local computer or *Select counters from computer* and then selecting the computer to monitor, which can be the local computer or another computer on the network. You can also specify the UNC for a computer, such as \\Lawyer (the \\ indicates a workstation or server). The server from which you start System Monitor is already inserted as the default computer. This is a powerful option for a server administrator, because it means you can monitor other network servers or workstations from one place. Many network administrators monitor a server from System Monitor in their Windows 2000 Professional workstation.

After selecting the computer to monitor, go to the Performance object list box and click the list arrow to select an object to monitor—such as Memory, in this example, to monitor paging. Make sure *Select counters from list* is selected, and for the counter, select Pages Input/sec (see Figure 14-10). This counter measures the number of virtual memory pages read back into memory per second. If you have a question about an object and counter combination, click the Explain button to view a description in a separate dialog box. After selecting the counter, click the Add button to start monitoring. While still in the Add Counters dialog box, select the options to monitor the object Memory and the counter Pages Output/sec, which shows the number of pages written to the page file each second. Also, select to monitor the % Usage and % Usage Peak counters for the Paging File object, to show the amount in use and the peak usage. The Paging File object also has instances that allow you to monitor a page file on a specific disk, if you use more than one page file or monitor all page files at once. Click Total as the instance, so that you monitor all page files at once. When you finish, click Close to view the four objects charted at the same time.

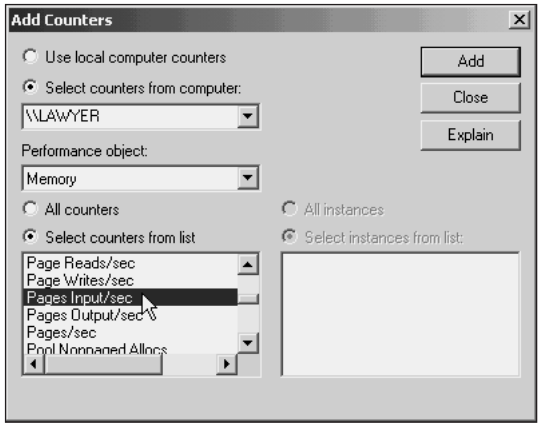

Figure 14-10 Selecting an object and its counter

Gather data for a short time before checking on the progress of System Monitor. Each object and counter combination is charted using a different color, so that one can be separated from the other. For instance, % Usage might be red, % Usage Peak might be green, while Pages Input/sec is blue, and Pages Output/sec is yellow (see Figure 14-11, which is not shown in color). The counters are shown at the bottom of the screen with a key to indicate the graphing color for each one. When you click one of these counters, the status bar just above the counters shows the following for that counter:

- *Last*—the current value of the monitored activity
- *Average*—the average value of the monitored activity for the elapsed time
- *Maximum*—the maximum value of the activity over the elapsed time
- *Minimum*—the minimum value of the activity over the elapsed time
- *Duration*—the amount of time to complete a full graph of the activity

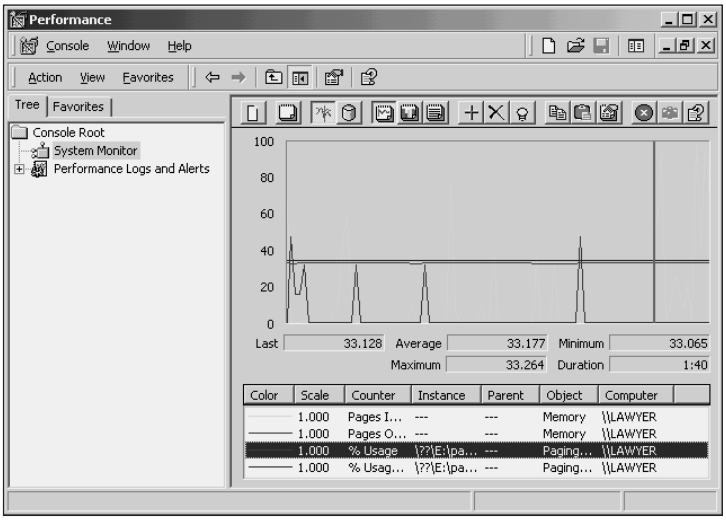

Figure 14-11 Monitoring page file performance

In Figure 14-11, the charting shows that paging looks good. The % Usage and the % Usage Peak are not over 40% at any point in time. But, some of the peaks for the Pages Input/sec are a concern. To follow up in this situation you might close the two paging-related counters, Pages Input/sec and Pages Output/sec by clicking each, one at a time, in the bottom of the screen, then clicking the Delete button.

In this example, you would further monitor the two counters for the Paging File object, % Usage and % Usage Peak, at a later time. You can save these settings and start and stop monitoring at different times. To save the settings, right-click the Copy Properties button, click Save As, and enter a file in which to store them. The next time you start System Monitor, open the file that you saved, copy its contents, and click the Paste Counter List button in System Monitor.

Assume that you have periodically monitored the Paging File counters % Usage and % Usage Peak over a period of a few days and find that the % Usage is almost always over 90% and that the % Usage Peak goes up to 100%. In addition, you monitor the Memory counter, Page Faults/sec, and continue checking the results. A **hard page fault** occurs when a program does not have enough physical memory to execute a given function. If there are frequently over five hard page faults per second, this is another strong indication of a memory bottleneck. In this situation, the combination of statistics indicates the need to add memory to the server.

Another situation resulting in a page fault occurs when two processes share the same 4 MB block of paged data. One process may read the block from disk into memory, just before the other process is about to do the same. The second process is unable to access the paged block, because it is in use. A page fault is also caused when there is not enough RAM to be shared by virtual memory and caching. All of these page fault problems are monitored by using Memory as the object and Page Faults/sec as the counter. One way to reduce page faults and improve performance is to increase RAM. This is especially important if database systems

such as Microsoft SQL Server or Oracle are installed, which are designed to share memory blocks when there is limited RAM.

A page fault in a kernel (operating system) process may occur if a reference to a page location is lost or corrupted. If this happens, the Windows 2000 system may crash with a Stop message. The error may be caused by a small power fluctuation, a damaged memory module, or a corrupted operating system file. Try rebooting to determine if the error recurs. If it does not, it was most likely caused by a transient situation, such as a power fluctuation. If it persists, test the computer's memory and replace damaged memory modules, or contact a Microsoft technician for information on how to read a crash dump to determine what process is linked to the crash.

Table 14-8 provides a summary of tips for monitoring and tuning memory and paging.

Table 14-8 Using System Monitor Objects and Counters to Monitor Memory and Paging

Object: Counter	Explanation
Memory: Available Bytes	Measures the bytes of memory available for use on the system. Microsoft recommends that this value be 4096 KB or higher. If values stay at or below this, your system will benefit from additional RAM. This figure is also available on the Task Manager Performance tab.
Memory: Cache Faults/sec	Measures the number of times the page file is called from disk or relocated in memory. Higher values indicate potential performance problems. (Higher values will be about double baseline values or more, on a lightly loaded system.) Remedy this by adding more memory; in this case, L2 cache (see Chapter 2) is better than adding main RAM.
Memory: Page Faults/sec	Returns a count of the average number of page faults per second for the current processor. Page faults occur whenever memory pages must be called from disk, which explains how memory overload can manifest as excessive disk activity. If the value is frequently over 5, or more than double that in a light-load baseline, consider adding more RAM.
Memory: Pages Input/sec Memory: Pages Output/sec	These counters measure the number of virtual memory pages read into (Input/sec) and out of (Output/sec) memory per second. If their total is frequently over 20, this shows a need to add RAM. By using both counters you can assess demands on memory and paging at once. Pages Input/sec translates into page faults. Pages Output/sec shows demand on memory, and when this value is frequently over 15–20, this indicates a need to add RAM.

Table 14-8 Using System Monitor Objects and Counters to Monitor Memory and Paging (continued)

Object: Counter	Explanation
Memory: Pages/sec	Tracks the number of pages written to or read from disk by the Virtual Memory Manager plus paging traffic for the system cache. If this value is more than double the light-load baseline, or if it is typically over 20, it indicates a need for additional RAM.
Paging File: % Usage Paging File: % Usage Peak	Both show how much of the page file is currently occupied. Neither object/counter should frequently exceed 99% but look at this information in relation to Memory:Pages Input/sec, Memory:Pages Output/sec, and Memory:Available Bytes. If the values are frequently over 99%, increase the page file size.
Server: Pool Paged Peak	Shows the most that the server has used in terms of virtual memory. This should be at least 1.5 times the size of RAM in the server.

Interaction Between Software and Memory Use

Software applications sometimes use the server's RAM very inefficiently, causing performance problems. Inefficient use of memory occurs for at least two reasons: poor program design and failure to return memory to the server after a process is complete.

Consider a program that needs to obtain four different data values from a database table. An inefficient program loads the database table, extracts the first value, and works on that value, going through the same process four times and therefore loading the entire table four times. A significant amount of memory is used each time the table is loaded. A more efficient way to design the program and use memory is to create in advance a view of the table that will include only the type of data that program accesses, such as the four data values. When the program needs the data, it loads only the view, which is a technique that uses much less memory than loading the entire table. The four data values are then extracted from the view at the same time, instead of in four different cycles. This design enables the program to access and work on the data faster, using much less memory and less processor time than the inefficient way of writing the program.

Another way in which programs use memory inefficiently is by **leaking memory**, which means that the application fails to release memory when it is no longer needed. This is a very common problem that has a cumulative impact, because the program may go through several cycles in which it repeatedly accesses blocks of memory that are not released. The result is that the page file continually grows, resulting in slower and slower server performance.

Adding RAM or increasing the page file size to combat the inefficiency of a program is not likely to address the server's performance problem. A better solution is to identify the program and redesign it or purchase one that is more efficient. System Monitor is an effective tool for identifying an inefficient program. In System Monitor, track the Process object and the counters, Page File Bytes and Page Faults/sec, for each process that you suspect is causing a problem. All of the currently running processes are listed in the instances list in System Monitor's Add Counters dialog box (see Figure 14-12). As you select processes to monitor,

also add Total as an instance to monitor the combined page faults for all processes. A high rate of page faults for one process is a strong indicator that there is a problem with that process.

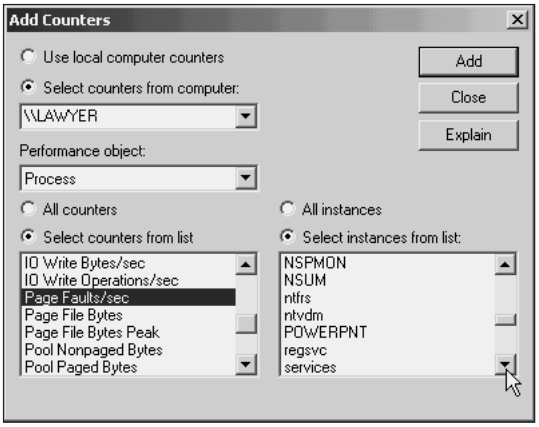

Figure 14-12 Using System Monitor to find an inefficient program process

Table 14-9 provides general tips for monitoring the relationship between software and memory.

Table 14-9 Using System Monitor Objects and Counters to Monitor Software Use of Memory

Object: Counter	Explanation
Process: Page Faults/sec	Measures the number of page faults for all threads in a process. Compare the number of page faults to the total amount of bytes for that process stored in the page file (Process:Page File Bytes). If the number of page faults is high and the number of bytes continues to grow, suspect a problem with leaking memory.
Process: Page File Bytes	Measures the number of bytes stored by a process in the page file. If this number continues to grow excessively as the program is running, suspect a problem with leaking memory.

Monitoring the Processor and Locating Processor Bottlenecks

Besides monitoring the processor in Task Manager, you can also use System Monitor, for example for diagnosing processor overload. There are three important components to studying the processor load:

- The percent of time the processor is in use
- The length of the queue containing processes waiting to run
- The frequency of interrupt requests from hardware

System Monitor has processor counters to measure each type of processor load. Start monitoring the processor load by selecting Processor as the object and % Processor Time as the counter. This counter measures how much the processor is in use at the present time. It is normal for the processor use to fluctuate between 50% to 100%. If the processor constantly remains at a high percentage, such as between 90% to 100%, this is an indicator that there is a problem.

When processor use is high, it is time to collect additional data by monitoring the number of processes waiting in line for their turn on the processor. Use the Processor Queue Length counter for the System object to determine if there is a queue of waiting processes. If the processor is often at 100%, but there are no processes waiting in the queue, the processor is handling the load. If four or five processes are always in line, this suggests that it is time to consider a faster processor.

You can monitor the same information about processor queue length by monitoring the Server Work Queues object and the Queue Length counter.

Before deciding that you need to purchase a new processor, make sure the processor load is not due to a malfunctioning hardware component, such as a NIC or disk adapter. When you monitor the processor load, add two additional counters, % Interrupt Time and Interrupts/sec, for the Processor object. A high frequency of interrupts per second, such as over 1000, is an indication that there is a problem with a hardware component. Also, frequently high % Interrupt Time, such as over 80%, is another indication of a hardware problem. These counters do not locate the component, but they do show that the overload problem is unlikely to be solved by a new processor. If you encounter a high level of hardware interrupts, check the system log (refer to Chapter 16) for information about hardware problems. Practice monitoring the processor in Hands-on Project 14-5.

Collect benchmarks on the level of hardware interrupts, so as to have comparative data for diagnosing problems later.

You can fine-tune your analysis of possible processor bottlenecks that are not related to a hardware problem by monitoring several additional objects and counters. For example, check for one or more processes that may be causing the load by monitoring two object:counter combinations: the Processor: % Processor Time and the Process: % Processor Time. You can select by name different processes to monitor as an instance. Also, in case the processor load is due to priorities set for certain processes, monitor the Process object using the Priority Base counter for each process (selected as the instance) that may be set too high. This method enables you to determine the exact priority of any process. If the bottleneck is due to a priority that is set too high, the solution to the bottleneck may be as simple as lowering the priority.

You can also use Task Manager to view all base priority class settings by accessing the Processes tab, clicking the View menu, clicking Select Columns, and clicking Base Priority.

When you determine that a bottleneck is focused on one process and that its priority is not set too high, monitor the threads used by that process by using the Thread object and % Processor Time as the counter. Each thread is displayed as an instance that you can select. For example, if the process FastBK looks like the culprit and has eight threads, monitor each thread from FastBK/1 to FastBK/8.

Sometimes the processor bottleneck is due to multiple processes running on the server. In this situation, determine how many threads within each process that are placing a load on the processor. For example, if the processes use an average of two or three threads, then the processing load is likely to be alleviated by upgrading to a faster processor. However, if the combined processes are using a high number of threads on average, such as 6 to 8 threads, then implementing a faster processor may not provide enough performance enhancement to solve the bottleneck. In this situation, it is necessary to upgrade to a multiple processor (SMP) computer, on which the processing load is equalized across processors. Table 14-10 gives tips on monitoring and tuning processor use.

Table 14-10 Using Objects and Counters to Monitor a Processor

Object: Counter	Explanation
Process: Priority Base	Measures the priority base of a selected process. This enables you to determine if one process is causing a processor bottleneck because its priority is set too high.
Processor: % Interrupt Time	Measures the amount of the processor's time that is used to service hardware requests from devices such as the NIC, disk and CD-ROM drives, and serial and parallel peripherals. A high rate of interrupts when compared to your baseline statistics indicates a possible hardware problem, such as a malfunctioning disk controller or NIC.
Processor: % Processor Time	Measures the percentage of time since System Monitor started that the CPU is busy handling nonidle threads. Sustained values of 80–85% or higher indicate a heavily loaded machine; consistent readings of 95% or higher may indicate a machine that needs to have its load reduced, or its capabilities increased (with a new machine, a motherboard upgrade, or a faster CPU).
Processor: Interrupts/sec	Measures the average number of times per second that the CPU is interrupted by devices requesting immediate processing. Network traffic and system clock activity establish a kind of background count with which this number should be compared. Problem levels occur when a malfunctioning device begins to generate spurious interrupts, or when excessive network traffic overwhelms a network adapter. In both cases, this will usually create a count that's five times, or greater, that of a lightly loaded baseline situation.

Table 14-10 Using Objects and Counters to Monitor a Processor (continued)

Object: Counter	Explanation
Server Work Queues: Queue Length	Indicates the number of items in a single processor's work queue. Frequent situations in which the queue length is over 4 indicate that the processor is experiencing a bottleneck.
System: Processor Queue Length	Measures the number of execution threads waiting for access to a CPU. If this value is frequently over 4 on a single CPU, it indicates a need to distribute this machine's load across other machines, or the need to increase its capabilities, usually by adding an additional CPU (where possible) or by upgrading the machine or the motherboard. When the value is over 2 per each CPU on multiple-processor systems, you should consider adding processors or increasing the processor speed.
Thread: % Processor Time	Measures the load on the processor due to threads running in processes. If 2 to 3 threads, on average, are running per each process, consider upgrading to a faster processor. If 6 to 8 threads are running, on average, per each process, consider upgrading the number of processors by using an SMP computer.

Monitoring Disk Performance and Disk Tuning

Before using System Monitor to study hard disk performance, it is necessary to install the Disk Performance Statistics Driver, which includes counters for disk monitoring. The driver is installed by running the program Diskperf, which is located in the \Winnt\System32 folder. Figure 14-13 shows the installation screen for the driver, which can be installed from the Command Prompt window or from the Start button, Run option. After the driver is installed, or after you change the driver configuration, the server needs to be rebooted. The parameters for Diskperf are as follows (try Hands-on Project 14-6):

- *Diskperf:* Indicates if the driver is installed and if the counters are started
- *Diskperf –y:* Installs the driver and the complete set of disk performance counters
- *Diskperf –yd:* Installs the driver and counters to work for physical drives
- *Diskperf –yv:* Installs the driver and counters to work for logical drives
- *Diskperf –n:* Deactivates the counters
- *Diskperf –nd:* Deactivates the counters only for physical drives
- *Diskperf –nv:* Deactivates the counters only for logical drives
- *Diskperf \\server:* Sets up the counters on the specified computer
- *Diskperf /?:* Displays the Diskperf parameter settings

Figure 14-13 Running Diskperf

After the Diskperf counters are loaded and you reboot the system, two new objects and their associated counters are available in System Monitor: LogicalDisk and PhysicalDisk. Use LogicalDisk to observe activity on a set of disks, such as a striped volume. Use PhysicalDisk if you want to monitor a specific disk, such as disk 0 in a set of five disks. Watch at least two counters, % Disk Time and Current Disk Queue Length. The first counter shows the amount of activity on a disk, and the second shows the number of waiting requests to access the disk. If one disk frequently is busy at the 100% level, information on the number of waiting requests helps to diagnose the problem. If there are 0 to 1 requests normally in the queue, the disk load is acceptable. If the queue generally has 2 or more requests, it is time to move some files from the overloaded disk to one less busy.

The best way to determine which files to move is to understand what applications and data are on the server and how they are used. If all of the server disks are constantly busy, it may be necessary to purchase disks with more spindles or to add additional data paths. (A spindle is a rod attached to the center of a hard disk platter and to a motor used to rotate the rod and disk.)

Individual drives typically have one spindle, and RAID drives have multiple spindles within the disk array. A RAID array is a good investment for growing servers because of the combined performance and redundancy features.

Another source of disk activity is the page file. Monitor the Memory counter Pages/sec and the PhysicalDisk counter % Disk Time simultaneously. This shows the paging activity in relationship to the activity on the disk. Sometimes the disk data transfer rate, which is measured by the PhysicalDisk counter Disk Bytes/sec, also is a problem. Use all three counters to track page file activity and how fast the page file is written to disk. This gives you a good idea of the page file activity and the disk speed at the same time, particularly since the page file is a very large file. Also, experiment with the disk transfer rate by copying a large file from floppy drive A to the hard disk you are monitoring. Another option is to monitor the transfer rate when a large number of records are written to a database on the disk. If page file activity is a problem, consider increasing RAM or implementing a page file on more than one disk. If

paging activity is low, but the transfer rate is slow for large files, such as the page file or a database file, consider upgrading to faster disks.

A visible indication that a disk may be a bottleneck is that its LED is lighting constantly and you can hear the disk busily reading and writing data. There are three general reasons why a disk is busy. One reason is simply that it is experiencing heavy sustained use. Heavy use is not an automatic indication that there is a problem, if the disk is handling the load. If the Current Disk Queue Length and Avg. Disk Queue Length generally stay in the 1 to 2 range, the disk is handling the load, even though you may see its lights on frequently. If the queue length is often in the 3 and over range, then you need to explore more about the problem, which leads to the other reasons why a disk is busy.

Another reason why a disk may appear as a bottleneck is that there is really a memory shortage causing disk activity because of heavy use by the page file. Use the techniques you have learned to monitor memory, paging, and file system cache to determine if there is a memory shortage. Also, monitor the following to check the link between paging and disk activity:

- The Memory object and Pages/sec counter
- The LogicalDisk object and the counters % Disk Time, Avg. Disk Queue Length, Avg. Disk Read Queue Length, and Avg. Disk Write Queue Length

Four additional reasons why one or more disks are busy or are a source of bottlenecks are:

- Fragmentation
- Disk fault tolerance method
- Location of files
- Disk speed

Perhaps the most common reason for a disk bottleneck is that one or more disks are heavily fragmented. You can easily address this problem by running the Disk Defragmenter, as explained in Chapter 7.

If you have configured disk storage RAID-5 volumes, the bottleneck may be linked to more active disk writing than you initially estimated. RAID-5 disks are able to read data faster than they can write it, because they must take time to calculate and write parity and fault tolerance data with each write operation. You can compare read to write activity by using the following System Monitor measures:

- For read activity, monitor the LogicalDisk and PhysicalDisk counters Avg. Disk Bytes/Read and Avg. Disk sec/Read.
- For write activity, monitor the LogicalDisk and PhysicalDisk counters Avg. Disk Bytes/Write and Avg. Disk sec/Write.

The Avg. Disk Bytes/Write and Avg. Disk Bytes/Read counters measure the average number of bytes transferred to or from the disk per each read or write activity. The Avg. Disksec/Read and Avg. Disksec/Write counters show the average number of seconds it takes to perform the disk read or write activity. If the disk write activity is much more frequent than read activity

and the users report delays in their work, consider using disk mirroring or duplexing instead of stripe sets with parity or RAID-5 volumes.

On disks that employ no fault tolerance measures or that are mirrored, the location of disk files can be important to diagnosing a bottleneck. Frequent visual inspection may show that one set of mirrored volumes is busier than another set. Suppose you have two sets of mirrored volumes, disk sets 0 and 1, and your visual inspections indicate that disk set 0 is often busy, but disk set 1 is not. You can study the discrepancy further by monitoring on both disks using the LogicalDisk counters % Disk Time, Avg. Disk Queue Length, Avg. Disk Read Queue Length, and Avg. Disk Write Queue Length. If you find, for example, that disk set 0 is nearly always busier than disk set 1, consider moving files. This tuning requires knowledge of the files and their purpose. For example, disk set 0 may contain a set of Microsoft Access databases that are used constantly, while disk set 1 has only sparse data. In this situation, you can spread the load of the databases between the two disks (consulting first with the users).

In some situations, a hard disk simply may have a slow transfer rate and may need to be upgraded, particularly if it is an older disk. As discussed earlier, measure the disk transfer rate by monitoring the PhysicalDisk counter Disk Bytes/sec along with % Disk Time. Set up a test by transferring large files to that disk or by developing a large query of a database. The disk performance is related to the data transfer rate of the disk adapter and controller, and the disk access time is the speed of the disk in accessing data. It can be very worthwhile to replace old disk technology with newer disks that use high-speed SCSI adapters and that have fast disk access times (see Chapter 2). Table 14-11 presents a summary of tips for monitoring and tuning disk performance.

Table 14-11 Using System Monitor Objects and Counters to Monitor Disk Performance

Object: Counter	Explanation
LogicalDisk: % Disk Time	Measures the percentage of time that a disk is busy with Read or Write requests. If this level is sustained at 80% or greater, redistribute files to spread the load across multiple logical drives. Also check the corresponding PhysicalDisk counter.
LogicalDisk: Avg. Disk Bytes/Read and LogicalDisk: Avg. Disk Bytes/Write	Used together, these provide a way to compare disk read to disk write activity, as a way to determine if you need to modify a currently established fault-tolerance method or add disk spindles.
LogicalDisk: Avg. Disk Bytes/Transfer	Measures the average number of bytes transferred between memory and disk during Read and Write operations. If the value is at or near 4 KB, this might mean excessive paging activity on that drive. A larger number indicates more efficient transfers than a smaller one, so watch for declines from the baseline as well.
LogicalDisk: Avg. Disk Queue Length and LogicalDisk: Current Disk Queue Length	These objects/counters indicate how many system requests are waiting for disk access. If the queue length is greater than 2 for any logical drive, consider redistributing the load across multiple logical disks, or if this is not possible, upgrade the disk subsystem. Also check the corresponding PhysicalDisk counters. Monitor these counters with Avg. Disk Read Queue Length and Avg. Disk Write Queue Length for more detailed statistics.

Table 14-11 Using System Monitor Objects and Counters to Monitor Disk Performance (continued)

Object: Counter	Explanation
PhysicalDisk: Avg. Disk Queue Length and PhysicalDisk: Current Disk Queue Length	These objects/counters track activity per hard disk, but provide much the same kind of information that the logical disk counters do. However, the problem threshold for physical disks is different than it is for logical ones. For physical disks, the threshold is between 1.5 and 2 times the number of spindles on the hard drive. For ordinary drives, this is the same as for logical disks. But for RAID arrays (which Windows 2000 treats as a single drive) the number is equal to 1.5 to 2 times the number of drives in the array. Monitor these counters with Avg. Disk Read Queue Length and Avg. Disk Write Queue Length for more detailed statistics.
PhysicalDisk: % Disk Time	Measures the percentage of time that a hard drive is kept busy handling Read or Write requests. The sustained average should not exceed 90%, but even if sustained averages are high, this value is not worrisome unless the corresponding queue length numbers are in the danger zone as well.
PhysicalDisk: Avg. Disk Bytes/Read and PhysicalDisk: Avg. Disk Bytes/Write	Used together, these provide a way to compare disk read to disk write activity, as a way to determine if you need to modify a currently established fault-tolerance method or add disk spindles.
PhysicalDisk: Avg. Disk Bytes/Transfer	Measures the average number of bytes transferred by Read or Write requests between the drive and memory. Here, smaller values are more worrisome than larger ones, because they can indicate inefficient use of drives and drive space. If a small value is caused by inefficient applications, try increasing file sizes. If it is caused by paging activity, an increase in RAM or cache memory is a good idea.
PhysicalDisk: Disk Bytes/sec	Tracks the number of bytes read from and written to disk each second. Use this object/counter combination to study the transfer rate of a disk to determine if you need to purchase a faster disk drive.

Monitoring Terminal Services

When you install terminal services on a server, you can monitor two objects through System Monitor: Terminal Services and Terminal Services Session. The Terminal Services object enables you to monitor active sessions, inactive sessions, and total sessions. The Terminal Services Session object enables you to view selected sessions or all sessions and to determine how they affect the server load—for example, by selecting Terminal Services Session as the object, selecting % Processor Time as the counter, and clicking to view all instances, as in Figure 14-14.

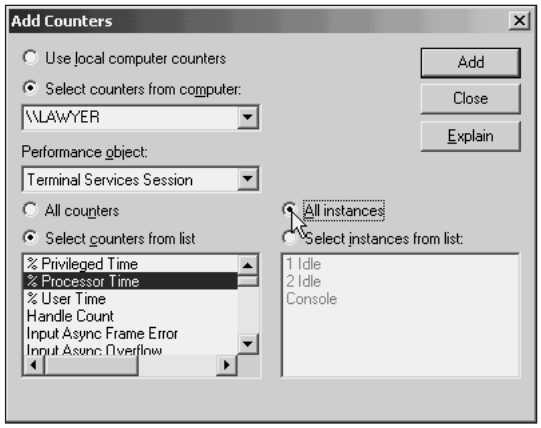

Figure 14-14 Monitoring terminal services

Monitoring File System Cache

Windows 2000 Server uses a portion of RAM for file system cache as a way to enhance a server's performance. **Cache** is employed by computer systems to store frequently used data in quickly accessed storage, such as memory. In Windows 2000 Server, file system cache operates as an intermediary between disk operations and an application that requests data. When the application requests data, the operating system first checks cache and then checks disk storage to locate the data.

In Windows 2000 Server, the operating system attempts to store both program code and data that have a high probability of being accessed, including code and data that have been used most recently, in the file system cache. In some cases, if the operating system determines that records in a particular file are likely to be accessed, it may cache the entire file.

When working with file systems, Windows 2000 does not truly load data into cache, but instead creates maps or pointers to the disk location of data. The pointers enable the system to quickly access data without engaging in a search process to obtain its disk location and then load the data. The cache pointers are kept in RAM only, relieving the stress on page file resources.

The success of file system caching is measured through cache hits and misses. A *cache hit* is an instance in which an application goes to cache and there is a pointer to the disk location of the data the application needs. A *cache miss* occurs when there is no pointer and a disk search process is used to find the data location. A low cache miss rate means that low disk I/O activity has been used to obtain data, because there was less searching to find the data. When the operating system determines there is no longer a need to store certain information in file system cache, it performs a *flush* to make that cache area available for the next cached information.

File system cache performance is influenced by the way in which an application obtains data and by the way data is stored in a file. The best performance results when an application accesses data sequentially and when data is stored sequentially in a file and on disk, because it

is not necessary to jump to different disk locations for data. A program's caching efficiency can be measured through System Monitor by watching the Cache object and the Copy Read Hits % counter. Microsoft recommends that this counter should be in the 80-90% range.

The Windows 2000 Server operating system controls the amount of RAM that is allocated to file system cache. Also, the amount allocated is determined by the amount of RAM in the server. For example, if there is 64 MB of RAM, the amount allocated to cache is likely to be about 10–15 MB, and for 128 MB RAM, cache will be around 30–40 MB. The Performance tab in Task Manager (refer to Figure 14-6) shows the amount of RAM allocated for file system cache.

Too little file system cache equates to a system bottleneck, particularly on servers in which software applications make extensive use of the cache. There are only limited ways to tune the cache on a server. The best way to tune cache is to install more memory. A second option is to increase the priority given to file system cache in memory, as you learned in Chapter 7. To increase the priority, open Network and Dial-up Connections, right-click Local Area Connection, and click Properties. Scroll the installed components list to find File and Printer Sharing for Microsoft Networks, and double-click that component to view memory tuning options. Click *Maximize data throughput for file sharing* to give a higher priority to file system caching, so more RAM is allocated by the operating system for this purpose (see Figure 14-15). If, instead, you click *Maximize data throughput for network applications*, priority (RAM space) is taken away from file system cache and used for working sets.

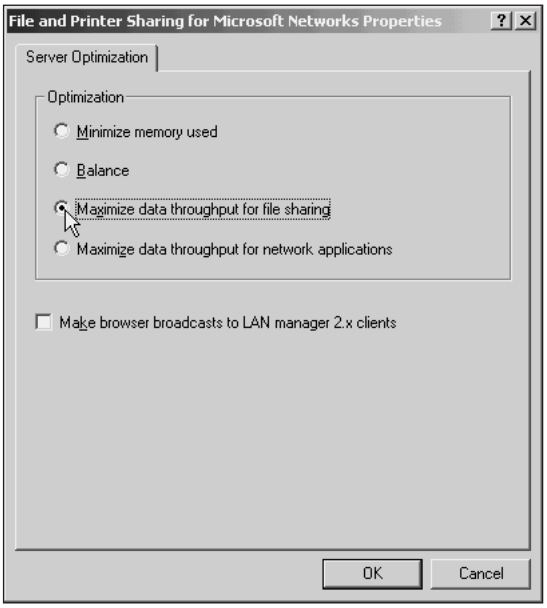

Figure 14-15 Tuning file system cache

A **working set** is the amount of RAM allocated to a running process. Thus, when you tune the server you need to determine if you need more RAM allocated to file system cache or more to running processes. One way is to use System Monitor to study cache needs for memory, compared to working set needs. To make a comparison, first monitor cache using the following at the same time: (1) Cache as the object and Copy Read Hits % as the counter, (2) Memory as the object and Available Bytes as the counter, and (3) Process as the object, Page Faults/sec as the counter, and Total as the instance. Next, monitor the working set activity by using at the same time: (1) Process as the object and the counters Working Set and Page Faults/sec, and (2) Memory as the object and Available Bytes as the counter. As a general rule, if the server is used most to run applications and access data files, consider tuning it to give priority to cache. If the server is mostly used to run processes, tune it for working sets. Table 14-12 presents tips for monitoring and tuning file system cache.

Table 14-12 Using System Monitor Objects and Counters to Monitor File System Cache

Object: Counter	Explanation
Cache: Copy Read Hits %	Tracks the file system cache access and should be in the range of 80–90%. If it is lower than this range, make sure the server is tuned to *Maximize data throughput for file sharing* or add more RAM.
Memory: Available Bytes	Measures the amount of RAM that can be used by processes
Process: Page Faults/sec	Measures the number of page faults for a process. Monitor the page faults of all processes by using Total as the instance.
Process: Working Set	Tracks the amount of RAM currently allocated to a process. Monitor using Total as the instance to determine how much RAM is allocated to all processes.

Using Performance Logs and Alerts

Performance logs and alerts work as partners with System Monitor. **Performance logs** are used to track performance data over a given period of time, and **alerts** are used to warn you of problems when they occur. There are two kinds of performance logs: counter logs and trace logs. A counter log traces information on System Monitor objects that you configure, taking a snapshot at intervals that you determine, such as every 15 seconds. Trace logs monitor particular events that you specify, so that the log contains only those instances when the events occur, for example creating a trace to record each time there is disk input/output activity or when there is an Active Directory Kerberos security event. After a log is created, you can open it from System Monitor to view its contents. You can also create a log in a format that can be imported into a spreadsheet, for example into Excel. Table 14-13 shows the file formats that are available.

Table 14-13 Counter Log File Formats

Format	Description
Text file – CSV (.csv extension)	Used to export data into a spreadsheet that employs comma delimiters after data lines
Text file – TSV (.tsv extension)	Used to export data into a spreadsheet that employs tab delimiters after data lines
Binary File (.blg extension)	Used when you want to stop and start performance recording
Binary Circular File (.blg extension)	Used when you want to record information for an extended time and automatically restart at the beginning of the file

In general, the steps to create a counter log are to open the Performance tool from the Administrative Tools menu, double-click Performance Logs and Alerts to display the objects under it in the tree, right-click Counter Logs, and click New Log Settings (see Figure 14-16). Enter a name for the log, click the Add button, and complete setting up the monitoring parameters in the Select Counters dialog box, which is similar to the Add Counters dialog box used in System Monitor setup. Try Hands-on Project 14-7 to set up a counter log.

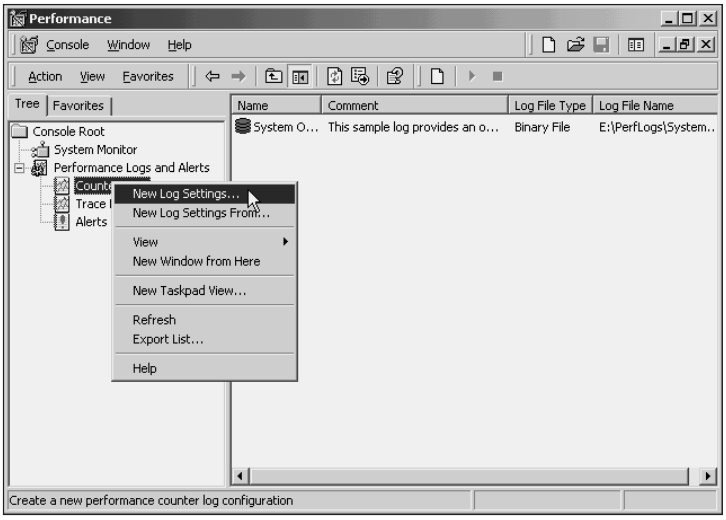

Figure 14-16 Configuring a counter log

Counter logs can occupy a significant amount of disk space and can slow system performance. Microsoft recommends that you take a snapshot at 15-second intervals or more frequently, if you plan to monitor for 4 hours or less. If you plan to monitor for over 4 hours, increase the interval. For example, if you monitor for 8 hours, take a snapshot at about every 5 minutes or so. Also, adjust the log file size so that it is large enough to hold the information sampled for a specified period of time.

If you need to manually stop the log before the specified stop time, right-click the log in the right-hand pane of the Performance tool, and then click Stop. You can restart the log by right-clicking it, and then clicking Start. Also, you can start a new logging session, for example, each morning, by right-clicking the log and then clicking Start. To add more objects and counters to monitor, right-click the log, click Properties, and then access the General tab. Click Add to use more objects and counters or click Remove to delete ones you do not want. Also, each new generation of a log (for instance if you start a log every morning for a week) is automatically labeled according to your specifications, which are set on the Log Files tab. Or, you can use the default, which is to end the saved log file from each recording session with an incremented number in the form "*nnnnnn*," such as 000001 for the first session, 000002 for next session, and so on.

Trace logs are useful when you do not want to continuously monitor performance, but want to document each instance of a particular event over a period of time. This is especially helpful in finding intermittent problems, such as excessive load on the server or network at certain times of the day or on certain days of the week. The elements that you can monitor in a trace log are more limited than those available to a counter log. Table 14-14 shows the trace log elements that can be monitored by the type of element. There are two types of elements: system provider and non-system-provider. The system provider elements are system processes, such as starting processes or performing disk operations. Non-system-processes are those handled by entities such as the Active Directory or local security provider. Also, there are only two file types available for trace logs: circular trace file and sequential trace file (both have a .etl extension). A circular trace file is used when you want the system to automatically overwrite data from the beginning of the file, when the file's capacity is exceeded. A sequential trace file is one in which the file is automatically closed when it is filled, and the system starts a new file. Hands-on Project 14-8 shows how to configure a trace log.

Table 14-14 Trace Log Elements That Can Be Monitored

System Process Elements to Monitor	Nonsystem Process Elements to Monitor
Process creations/deletions	Active Directory: Kerberos
Thread creations/deletions	Active Directory: SAM
Disk input/output	Active Directory: NetLogon
Network TCP/IP	Local Security Authority (LSA)
Page faults	Windows NT Active Directory Service
File details	

An alert gives you immediate warning when a problem occurs. For example, you may want Windows 2000 Server to send you an alert each time that the CPU is at 100% utilization. Each alert is sent to specific accounts, such as to the accounts belonging to the Administrators domain local group. To configure an alert to track 100% CPU use, for example:

1. Open the Performance tool from the Administrative Tools menu.
2. Double-click Performance Logs and Alerts to display the objects under it in the tree.

3. Right-click Alerts and click New Alert Settings.
4. Enter a name for the alert and click OK.
5. Click Add.
6. Select the option to monitor the Processor as the object, % Processor Time as the counter, and select the instance, such as Total, to monitor all processors. Click Add and then click Close. Click OK if a warning is displayed that you must set the alert limit.
7. Make sure that the object and counter are highlighted, and then enter Over in the *Alert when the value is* box. Enter 99 in the *Limit box*.
8. Set the interval at which the system should check for this event, such as every 5 or 10 seconds.
9. Click the Action tab. Check *Log an entry in the application event log*, if it is not already checked. Check *Send a network message to* and specify your account name or a group, such as Domain Admins. There also are options to start a performance log or to run a program when the event occurs.
10. Click OK.
11. Click Alerts in the tree, and then right-click the Alert that you created and click Properties, if you want to modify any of the parameters that you configured.

System Monitor, performance logging, and alert monitoring have been enhanced in Windows 2000 to use less CPU and memory resources, but they do occupy some resources. Use these tools only when you need them so that you do not regularly affect server performance with monitoring activities.

Chapter Summary

- The best preventive medicine for keeping server performance at its peak is to develop solid monitoring techniques. The first step in the monitoring process is to create a set of benchmarks so you have a way to compare normal performance to situations in which there are performance, hardware, or software problems. Benchmarks enable you to quickly identify and address problem areas as they develop.
- Server services are often the first place administrators go to monitor Windows 2000 Server, because so many critical functions rely on these services working smoothly. Other areas to monitor are the users who are logged on and the resources they access, such as shared folders.
- One of the easiest-to-use monitoring tools is Task Manager. Through it you can stop applications that are hung, tune processes, and keep track of memory and CPU use. Tuning a process can be an inexpensive and effective way to improve server performance.

- System Monitor is a versatile and widely used Windows 2000 Server monitoring tool. It enables you to monitor key server components such as processor, disk, and memory performance. There are hundreds of System Monitor configurations to track and diagnose almost every type of server problem. Performance logs and alerts employ System Monitor capabilities to enable you to track information in logs for later reference and to receive instantaneous notification of when there is a problem.

In the next chapter, you learn how to implement and use network monitoring tools to help in tuning network performance and diagnosing problems.

KEY TERMS

alert — Provides a warning of a specific Windows 2000 Server system or network event. The warning is sent to designated users.

backup browser — A computer in a domain or workgroup that maintains a static list of domain/workgroup resources to provide to clients browsing the network. The backup browser periodically receives updates to the browse list from the master browser.

base priority class — The initial priority assigned to a program process or thread in the program code by Windows 2000 when the program is started.

benchmark — A measurement standard for hardware or software used to establish performance baselines under varying loads or circumstances. Also called a baseline.

cache — Storage used by a computer system to house frequently used data in quickly accessed storage, such as memory.

counter — Used by System Monitor, this is a measurement technique for an object, for example, for measuring the processor performance by percentage in use.

file lock — Flagging a file so that it cannot be updated by more than one user at a time, giving the first user to access it the ability to perform an update.

handle — A resource, such as a file, used by a program that has its own identification so the program is able to access it.

hard page fault — When a program does not have enough physical memory to execute a given function and must obtain information from disk.

instance — Used by System Monitor, when there are two or more types of elements to monitor, such as two or more threads or disk drives.

leaking memory — Failing to return memory for general use after a process is finished using a specific memory block.

master browser — On a Microsoft network, the computer designated to keep the main list of logged-on computers.

named pipes — A communications link between two processes, which may be local to the server or remote, for example, between the server and a workstation.

performance log — Tracks system and network performance information in a log that can be viewed later or imported into a spreadsheet, such as Microsoft Excel.

process — An executable program that is currently running, such as Microsoft Word. A process may launch additional processes that are linked to it, such as a Help process to view documentation or a search process to find a file.

process tree — All of the processes that run directly or indirectly in association with an original process.

System Monitor — The Windows 2000 utility used to track system or application objects. For each object type there are one or more counters that can be logged for later analysis, or tracked in real time for immediate system monitoring.

thread — A block of program code executing within a running process. One process may launch one or more threads.

working set — Amount of RAM allocated to a running process.

Review Questions

1. Your advertising firm is expecting a new client to visit in about 15 minutes. In preparation, you have been printing out reports and graphics for the meeting, but the print process has been going slowly because the server is so busy. What can you do to best help ensure that the printouts are finished in time?
   a. Log all other users off, even if there is not time to give them sufficient notification.
   b. Increase the priority of the print spooler process to AboveNormal or High.
   c. Decrease the priority of all processes, except the spooler process to Low.
   d. Quickly increase the page file size by 1–2 MB to handle the printouts.
2. You are about to back up a Windows 2000 server in a small office and everyone has gone home for the evening. However, you notice through a window in a locked office that a computer is still logged on and seems to have a file open. How can you log off that computer from the server?
   a. Use the System Tools option in the Computer Management console.
   b. Use the Services and Applications option in the Computer Management console.
   c. Right-click that user's account in Local Users and Groups and access the General tab.
   d. Right-click that user's account in Local Users and Groups and click Log Off.
3. You are using the System Monitor Memory object and the Available Bytes counter to monitor memory. Which of the following results would be an indication that you need to add more RAM?
   a. frequent values between 7 and 10 MB
   b. frequent values of 3 MB or lower
   c. frequent values of 1 to 2 MB
   d. all of the above
   e. only a and b
   f. only b and c

4. Your Windows 2000 server is running slowly, and you suspect there is a program or program process that is causing the problem because you just installed eight new programs on it which are run by you on the server and by clients using terminal services. Which of the following tools enable you to monitor for the problem?
   a. Task Manager using the Processes tab
   b. System Monitor using the Process object
   c. Computer Management tool using the Services option
   d. all of the above
   e. only a and b
   f. only b and c
5. Which of the following modes can you use in System Monitor for displaying tracked data?
   a. histogram
   b. chart
   c. report
   d. all of the above
   e. only a and b
   f. only b and c
6. Several of your Windows 2000 Server clients are running Windows 3.11 and may be contending as master browsers. How can you monitor possible contention?
   a. Monitor the processes in Task Manager.
   b. Use the System Monitor Browser object.
   c. Monitor the number of user connections and the resources they are employing.
   d. Monitor for excessive cache hits.
7. While you are directly logged on to a Windows 2000 server using Internet Explorer, you discover that the program is not responding to any keystrokes you make, including your attempts to close the program. When you go to another window, all functions are normal. How can you close Internet Explorer?
   a. Use the Services tool to stop and restart the Explorer service.
   b. Use the Computer Management tool to stop and restart the Browser service.
   c. Use the Performance snap-in to set the priority of the Explorer.exe program to 0.
   d. Use Task Manager to end the Internet Explorer task.
8. You are receiving calls from users saying that they cannot log on to a Windows 2000 server over the network. You know that the server has a reliable NIC and network connection and that it is running without apparent problems. Which service should you check as one place to start working on the problem?
   a. Workstation
   b. System Event Notification

c. Plug and Play

d. RunAs

9. Through monitoring with System Monitor, you have located a program that is slowing server response because it is leaking memory. What is the best solution?

   a. Have the vendor fix the program or purchase another one that does not leak memory.

   b. Add more RAM.

   c. Install a faster processor.

   d. Increase the page file size.

10. You want to tune file caching on your server, but before you do, you decide to monitor file caching. Which of the following monitoring tools will help provide the performance information you need before tuning the server?

    a. System Monitor, using the Processor object and % Processor counter

    b. Task Manager, using the Performance tab

    c. System Monitor, using the Cache object and Copy Read Hits % counter

    d. all of the above

    e. only a and c

    f. only b and c

11. You are gathering system performance statistics, using the counter log, so that you can compile them into a spreadsheet to present to the Computer Resources Committee in your organization. What file type can you employ to store the data using tab delimiters for importing into a spreadsheet?

    a. text file format, using the .csv extension

    b. binary circular file format, using the .blg extension

    c. binary file format, using the .blg extension

    d. text file format, using the .tsv extension

12. You have assigned your assistant to monitor paging on the server to determine if more RAM is needed. In the absence of baselines, what number of page faults per second should she look for that would indicate it may be time to add RAM?

    a. more than 5

    b. 3 to 4

    c. 2 to 3

    d. 1 to 2

13. You work for a company that has servers distributed in different locations throughout a business building. How can you monitor, start, and stop server services from a Windows 2000 server located next to your office, so that you do not have to walk all over the building?
    a. Use System Monitor.
    b. Use the MMC Performance snap-in.
    c. Use the MMC Services snap-in.
    d. Services cannot be monitored on another server, except by using terminal services.
14. Which of the following may be solutions to relieving a processor bottleneck?
    a. Lower the priority of a particular process.
    b. Replace mirrored volumes with RAID-5 volumes as a way to reduce the number of CPU calls from adapters.
    c. Locate a possible hardware problem in which the device constantly accesses the processor.
    d. all of the above
    e. none of the above
    f. only a and b
    g. only a and c
15. While practicing, your assistant changed the priority of Windows Explorer, and now the server response for all users is extremely slow. What priority did he most likely set?
    a. High
    b. Normal
    c. Realtime
    d. Low
16. In Question 15, if you did not already know the process for which the priority was changed, which tool could you use to determine the process?
    a. Task Manager Performance tab
    b. Task Manager Processes tab
    c. System Monitor, using the Process object
    d. all of the above
    e. only a and b
    f. only b and c
17. Your boss has been using System Monitor to exclusively track the Processor object and % Processor time on a single CPU server. She does not have much time to monitor, but has decided that it is time to purchase a faster CPU because % Processor time occasionally reaches 85 to 100%. What is your advice?
    a. You agree that it is time to purchase a new CPU.
    b. You recommend adding more RAM instead.

c. You recommend rebooting the server immediately, because this is a sign of CPU leakage.

d. You recommend doing nothing, because this only indicates efficient use of the CPU by some programs and services.

18. One of your server administrator colleagues in another company is struggling to understand the data produced by System Monitor. The servers and network at that company have been in place for about 5 months, but he has been too busy setting up clients to gather information about server performance. Now he does not know how to interpret the information so as to determine in what areas performance is normal and in what areas it is not. What should he have done in advance?

    a. Gather server and network benchmarks.

    b. Started the System Monitor from day one and left it running continuously to gather data on the 10 most critical monitor objects.

    c. Run a trace log at least two full days a week, every week, monitoring system provider events.

    d. all of the above

    e. only a and c

    f. only b and c

19. You want to stop the Remote Procedure Call service, but are not sure what other services depend on it. How can you most easily find out?

    a. View the properties for the Remote Procedure Call service in the Services tool and access the Dependencies tab.

    b. Open the Services Dependencies object in the tree of the Services tool.

    c. Stop the services that you think it might depend on and look for the error messages.

    d. Look in the System Information option under the console tree in the Computer Management tool.

20. You want to set up a counter log to track system activity over a 12-hour period on two processor-based counters and one memory counter. What adjustments should you make when you set up the counter log?

    a. Set the log file size large enough to hold this much information.

    b. Only track system provider events for those objects.

    c. Set the sampling interval relatively high, for example, once every 10 or 20 minutes.

    d. all of the above

    e. only a and b

    f. only a and c

21. Which of the following is not an object that you might monitor in the System Monitor?

    a. Job Object

    b. Distributed Transaction Coordinator

    c. Network Connection Linker

    d. Server Work Queues

22. The vice president of marketing is calling you because he is trying to update an Excel spreadsheet that is shared from a Windows 2000 server. On the phone he is concerned because when he enters new data, there is a message that the file cannot be updated. Which of the following might cause this problem and what tool can you use to find the cause?
    a. The share is hidden, which means it cannot be updated. Use the Computer Management tool to check.
    b. The file is locked by another user. Use the Computer Management tool to check if this is the case.
    c. The Excel.exe process priority is set too low, at BelowNormal, which means updates are not processed. Use Task Manager to change the process's priority.
    d. all of the above
    e. only a and b
    f. only a and c
23. Your assistant wants to monitor the number of clients actively using a Windows 2000 terminal server. How can this be accomplished using System Monitor?
    a. Monitor the Server object
    b. Monitor the Terminal Services object
    c. Monitor the Browser object
    d. Terminal Services clients can only be monitored using the Terminal Services Manager, which is not part of System Monitor.
24. Which of the following System Monitor objects would you monitor to study paging activity on a server?
    a. Paging File
    b. Memory
    c. Server
    d. all of the above
    e. only a and b
    f. only a and c
25. Which of the following should you monitor when watching for a processor bottleneck in Windows 2000 Server?
    a. The length of the queue that holds processes waiting to access the CPU.
    b. The typical number of interrupt requests from hardware devices.
    c. The percentage of time in which the processor is engaged in non-idle processes.
    d. all of the above
    e. none of the above
    f. only a and b
    g. only a and c

HANDS-ON PROJECTS

Project 14-1

In this project you practice monitoring, starting, and stopping a service. You also view the service's dependencies.

To monitor the service, manage it, and view its dependencies:

1. Click **Start**, point to **Programs**, point to **Administrative Tools**, and click **Services**.
2. Scroll the right-hand pane to view the services that are installed and then find the ClipBook service.
3. What is the status and Startup Type of the service? Record your observations in your lab journal or in a word-processed document.
4. Double-click the **ClipBook** service.
5. If the service is not started, click the **Start** button, or if it is started, click the **Stop** button.
6. If you started the service, now click the **Stop** button, or if you stopped it, click the **Start** button.
7. How would you change the startup type? Record your observations.
8. Click each tab to view what it does and note your observations.
9. When you click the Dependencies tab, what services are dependent on the ClipBook service? On what services does the ClipBook service depend? Record your observations.
10. Click **OK** and then close the Services tool.

Project 14-2

In this project, you practice viewing the users who are connected to a Windows 2000 server and determining which files are locked.

To view the user connections:

1. Right-click **My Computer** on the desktop and click **Manage**.
2. Double-click **Shared Folders** under System Tools in the tree.
3. Click **Sessions** in the tree.
4. How many users are connected to the server? How many of those users have open files? Record your observations.
5. Click **Open Files** in the tree.
6. Are there locked files? If so, record some examples of files that are locked.
7. Record how you would view the shares that are set up on a server and the number of clients connected to each share.
8. Close the **Computer Management** tool.

Project 14-3

In this activity you set a process's priority and stop a task. To perform this activity, open Control Panel and then My Computer before starting.

To set the priority and stop a task:

1. Press **Ctrl+Alt+Del**.
2. Click the **Task Manager** button in the Windows 2000 Security dialog box.
3. Click the **Applications** tab, if it is not already displayed. What applications are currently running?
4. Right-click **My Computer** and then click **Go To Process**. (This takes you to the Processes tab.)
5. Right-click the **explorer.exe** process.
6. Click **Set Priority** and then click **AboveNormal**. Click **Yes** in the Task Manager Warning dialog box. What might happen if you clicked Realtime instead of AboveNormal? Record your answer.
7. Click the **Applications** tab.
8. Click **Control Panel** and then the **End Task** button. What happens to Control Panel? Record your observation.
9. Close Task Manager.

Project 14-4

This project gives you an opportunity to practice viewing objects, counters, and instances in System Monitor.

To view System Monitor objects, counters, and instances:

1. Click **Start**, point to **Programs**, point to **Administrative Tools**, and click **Performance**.
2. Double-click **Console Root** to view System Monitor, if it is not displayed.
3. Click **System Monitor**.
4. What is displayed in the right-hand pane?
5. Click the **Add** button (a plus sign) in the button bar in the right-hand pane.
6. What computer is selected by default for monitoring? How would you monitor activity on a different computer? Record your observations.
7. Click the down arrow in the Performance object box. Scroll through the options and record some of them in your lab journal or in a word-processed document.
8. Select the default, which is **Processor**. Make sure that the **Select counters from list** radio button is selected. Scroll to view the counters associated with the Processor object. What instances are available?
9. Next, click **Server** as the object. How many counters and instances are associated with this object?

10. Click **Process** as the object. How many counters and instances are associated with the Process object?
11. Observe two more objects and their associated counters and instances.
12. Click **Close** in the Add Counters dialog box, but leave the Performance console open for the next assignments.

Project 14-5

In this project, you use System Monitor to check for processor bottlenecks, such as the processor's ability to handle the server load and possible problems caused by hardware.

To check for processor bottlenecks:

1. Make sure that the Performance console is already open; if not, open it to display System Monitor.
2. Click the **Add** button in the button bar to add counters.
3. If it is not already selected, click **Select counters from computer** and make sure that the computer you are using is selected.
4. Make sure that **Processor** is selected in the Performance object box.
5. Make sure that **Select counters from list** is selected and click **% Processor Time**. Leave **Total** as the default for instances. What information does this counter provide for the Processor object? How would you find out, if you didn't know? Record your observations.
6. Click **Add**.
7. Click **% Interrupt Time** as the counter and leave **Total** as the instance. Click **Add**.
8. Scroll the counters list and click **Interrupts/sec**. Leave **Total** as the instance and click **Add**. How do the % Interrupt Time and Interrupts/sec counters help in diagnosing a processor bottleneck?
9. In the Performance object box, select **System**.
10. Scroll the counters list and click **Processor Queue Length**. Are there any instances? Click **Add**. What information does this object and counter combination provide in monitoring for a processor bottleneck?
11. Click **Close**.
12. Monitor the system for several minutes to determine if there are any processor problems. Record any problems that you diagnose using System Monitor. How would you change the view mode from a chart to a histogram? How would you add another object to monitor? Record your observations.
13. In the bottom portion of the right-hand pane, click each counter, one at a time, and click the **Delete** button (an X) on the button bar. (Leave the Performance console open for Hands-on Projects 14-7 and 14-8.)

Project 14-6

In this activity, you check on whether the Diskperf driver is installed and the System Monitor disk counters are activated:

To check on Diskperf's status:

1. Make sure you are logged on as Administrator or with Administrator privileges.
2. Click **Start**, point to **Programs**, point to **Accessories**, and click **Command Prompt**.
3. In the Command Prompt window, type **diskperf**.
4. What message is displayed in the first line after the command? What other information is provided? Record your results.
5. How would you start Diskperf to set up counters only for physical drives?
6. Type **exit** and press **Enter** to close the Command Prompt window.

Project 14-7

This project gives you practice in creating a counter log. Consider a situation in which you want to monitor paging and memory for a typical workday to help determine if you need to adjust the page file size or to add more RAM.

To set up a counter log for monitoring performance over a period of 8 hours:

1. Make sure that the Performance console is already open, and if not, open it to display Performance Logs and Alerts.
2. Double-click **Performance Logs and Alerts** to display the objects under it in the tree.
3. Right-click **Counter Logs** and click **New Log Settings** (refer to Figure 14-16).
4. Enter **Mem** and your initials, such as **MemMJP**, as a name for the log and then click **OK**.
5. Click the **Add** button.
6. Compare the options that you see to those you can use in System Monitor and record your observations.
7. Click **Use local computer counters**.
8. Choose the performance object, **Paging File**, select **% Usage** as the counter, and then select **_Total** as the instance. Click **Add**.
9. Choose the object, **Paging File**, select **% Usage Peak** as the counter, and then select **_Total** as the instance. Click **Add**.
10. Choose the object **Memory**, select **Page Faults/sec** as the counter. Are there any instances from which to select? Click **Add**.
11. Click **Close**.
12. Set the Interval box to **5** and the Units box to **minutes**.
13. Click **OK**.
14. Click **Yes** to create the folder for the log in which to record the information, if a folder has not already been created.

15. Click **Counter Logs** in the tree. There will be a green disk icon in front of the log name in the right-hand pane that shows the log is active and gathering data. Right-click the log, click **Properties**, and then click the **Schedule** tab.
16. In the Start log section of the dialog box, make sure that **At** is clicked and set the start time and date to match the current time and date.
17. Next, in the Stop log section, click **After** and enter **8** in the After box and **hours** in the Units box.
18. Click the **Log Files** tab. List the options for the Log file type and record them. (Note that any of the file formats can be converted to another format by using the Save As option when you right-click the log in the right-hand pane.)
19. Click **OK**. How would you view the log's contents?
20. After the log has run for a few minutes, right-click it and then click **Delete**.

Project 14-8

In this project, you set up a trace log to monitor each time a page fault occurs. Because this type of monitoring requires extra system resources, you will only set it up to monitor for 30 minutes.

To create a trace log of page faults:

1. Make sure that the Performance console is already open, and if not, open it to display Performance Logs and Alerts. Make sure that Trace Logs is displayed under it in the tree.
2. Right-click **Trace Logs** and then click **New Log Settings**.
3. Enter **CPU** and your initials as the log name, such as **CPUMJP**. Click **OK**. (If no folder has previously been created for log files, click OK to create the folder.)
4. On the General tab, click **Events logged by system provider** and make sure that only **Process creations/deletions** and **Page faults** are checked. Notice the name and location of the log file, which is where you can access it later to obtain its contents. Record this information.
5. Click the **Schedule** tab.
6. If it is not already selected, click the **Manually (using the shortcut menu)** radio button to start the log manually.
7. In the Stop log area, click **After**, and set the After box to **30** and the Units box to **minutes**.
8. Click **OK**. (Click **Yes** if you are asked to create the \Perf Logs folder.)
9. Click **Trace logs** in the tree and find the log you created in the right pane. What color is the icon that represents the log? Right-click the log and make sure that the Start option is deactivated, which means that it is already started. If it is not started, click **Start**.
10. Instead of waiting for 30 minutes, manually stop the log file after about 10 minutes, by right-clicking the log name, such as CPUMJP, in the right-hand pane and then clicking **Stop**. How would you restart the log?
11. Close the Performance console.

CASE PROJECTS

Aspen Consulting Project: Server Monitoring

Funds Unlimited is a firm that nonprofit organizations hire to help plan fundraising strategies and projects. The firm handles clients such as colleges, universities, charitable organizations, and others. They have two Windows 2000 servers that provide networked services to 80 consultants, managers, and staff members. The servers and network have been in use for about one year and now there seem to be some performance problems that Funds Unlimited wants to address. They have hired you to work with four staff members who administer the servers.

1. Funds Unlimited has never taken the time to gather server performance benchmarks. Develop a plan for gathering benchmarks that will help them monitor the server, perform regular tuning maintenance, and diagnose problems.
2. The server administrators are very unfamiliar with the basic server monitoring tools. Prepare a brief description of each of the following tools, including how to access each tool:
   - Services MMC snap-in
   - Computer Management tool and the Shared Folders option
   - Task Manager
   - System Monitor
   - Performance logs and alerts
3. While you are on-site, one of the programs that is running on the server stops responding. Explain how to close the program. While you are closing the program, also explain to the server administrators how to set the priority on a process, including any precautions you have to take when doing this.
4. Funds Unlimited has recently implemented a new client/server system in which a large database is kept on one of the servers and the program files are on the other server. The database server experiences frequent slowdowns throughout the day, but no one has kept track of specific instances. The management is discussing whether to purchase a faster CPU or an SMP computer to speed up access to the database server. Explain how they might gather information about the server's performance before making a decision to upgrade the server. In particular, address how they might do the following:
   - Monitor the number of users
   - Monitor the processor
   - Monitor memory and paging
   - Monitor processes used by the client/server system
   - Monitor disk response
5. Explain how to set up a counter log to assist with the monitoring that you described in Assignment 4. Also, explain how they might set up alerts to help in gathering information.

Optional Case Projects for Teams

Team Case One

There are many ways to analyze memory, paging, and processor interaction on a Windows 2000 server. Mark Arnez asks you to form a group to develop a set of guidelines that can be used to learn when to add more RAM to a server and when to upgrade the processor.

Team Case Two

Mark Arnez asks your same team to work on a document that shows how to develop a plan for establishing server benchmarks in an organization. Develop two models as examples of how to develop benchmarks. For one model, use a small office that has one server and that uses mainly word-processing, spreadsheet, and small database applications. In the second model, use a larger business in which there are 10–40 servers, many of which are running databases, client/server software, multimedia software, and graphics/publishing software.